

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
22 septembre 2005 (22.09.2005)

PCT

(10) Numéro de publication internationale
WO 2005/088927 A1

(51) Classification internationale des brevets⁷ : **H04L 29/06**

Cédric [FR/FR]; 5, rue Monseigneur Duchesne, F-35000
Rennes (FR).

(21) Numéro de la demande internationale :
PCT/FR2004/000337

(74) Mandataires : **LOUISET, Raphaël** etc.; Cabinet Plasser-
aud, 65/67, rue de la Victoire, F-75440 Paris Cedex 09
(FR).

(22) Date de dépôt international :
13 février 2004 (13.02.2004)

(81) États désignés (sauf indication contraire, pour tout titre de
protection nationale disponible) : AE, AG, AL, AM, AT,
AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO,
CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB,
GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG,
KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG,
MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH,
PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(25) Langue de dépôt : français

(26) Langue de publication : français

(71) Déposant (pour tous les États désignés sauf US) :
FRANCE TELECOM [FR/FR]; 6, place d'Alleray,
F-75015 Paris (FR).

(72) Inventeur; et

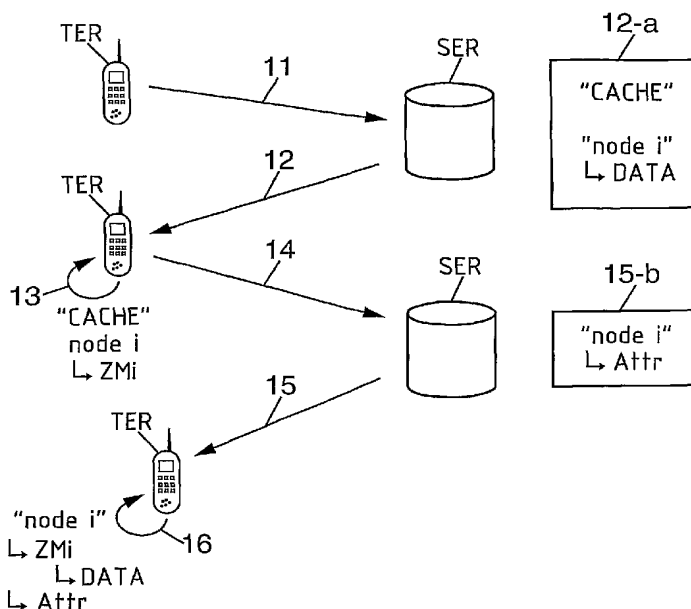
(84) États désignés (sauf indication contraire, pour tout titre de
protection régionale disponible) : ARIPO (BW, GH, GM,

(75) Inventeur/Déposant (pour US seulement) : **GEGOUT,**

[Suite sur la page suivante]

(54) Title: METHOD FOR EDITING A MULTIMEDIA PAGE USING PRE-STORED DATA

(54) Titre : PROCEDE D'EDITION D'UNE PAGE MULTIMEDIA AVEC PRE-MEMORISATION



(57) Abstract: The invention relates to multimedia page editing on a terminal. A server (SER) supplies pages to the terminal (TER) in the form of object arrangement commands for objects in a multimedia page to be generated. The method according to the invention comprises (a) a preliminary step (12) wherein the server transmits an object-related data packet as well as a temporary store command (CACHE) for the data (DATA), which is identified by a link (node i) in a terminal memory; and (b) a main step (15) wherein the server transmits a descriptive file containing said link for editing the object in a scene being generated in the terminal on the basis of the stored data.

[Suite sur la page suivante]

WO 2005/088927 A1



KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasién (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclaration en vertu de la règle 4.17 :

- *relative à la qualité d'inventeur (règle 4.17.iv)) pour US seulement*

Publiée :

- *avec rapport de recherche internationale*

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : L'invention concerne l'édition de pages multimédia auprès d'un terminal. Un serveur (SER) distribue des pages sous forme d'instructions d'agencements d'objets au sein d'une page multimédia à construire, auprès d'un terminal (TER). Le procédé au sens de l'invention comporte a) une étape préalable (12) pendant laquelle le serveur transmet un paquet de données relatives à un objet, avec une instruction de stockage temporaire ("CACHE") de ces données (DATA), identifiées par un lien ("node i"), dans une mémoire du terminal, b) et une étape courante (15) pendant laquelle le serveur transmet un fichier descriptif comportant ce lien, pour éditer, auprès du terminal, l'objet dans une scène en construction, à partir des données stockées.

PROCEDE D' EDITION D' UNE PAGE MULTIMEDIA AVEC PRE-MEMORISATION

- 5 L'invention concerne l'édition de pages multimédia auprès de terminaux, notamment dans le cadre de services multimédia proposés sur des téléphones mobiles agencés pour coopérer avec des réseaux cellulaires.
- 10 Dans le contexte de l'invention, un serveur distribue, à un ou plusieurs terminaux, une partie au moins de pages multimédia sous forme d'instructions d'agencements d'objets au sein d'une page multimédia à construire.
- 15 Dans ce contexte, un même objet peut être utilisé notamment par plusieurs pages multimédia successives et conserver, ou non, de mêmes paramètres d'agencements d'une page multimédia à l'autre.
- 20 On entend par "*page multimédia*" par exemple une scène graphique à éditer auprès du terminal, le cas échéant agrémentée d'une ou plusieurs séquences sonores à jouer sur des baffles ou écouteurs du terminal.
- 25 Par exemple, pour l'édition graphique d'un plan d'une ville sur un terminal mobile, un même objet (par exemple un objet graphique tel qu'une carte routière) est susceptible d'être édité dans plusieurs scènes successives si l'utilisateur du terminal applique une commande
- 30 interactive du type zoom/dézoom ou "flèche gauche/flèche droite, haut/bas" pour ajuster une position de

visualisation du plan. Dans ce cas, à chaque commande d'interaction de l'utilisateur, le serveur transmet habituellement les mêmes données relatives à l'objet graphique à éditer, le cas échéant avec des paramètres d'agencement de l'objet dans la scène graphique (par exemple une valeur de zoom), qui sont différents.

Il se pose alors le problème de transmission et stockage systématiques et inutiles des données relatives à un même objet, pour plusieurs pages multimédia dans lesquelles le même objet intervient, le cas échéant avec des paramètres d'agencement différents. Ce problème devient particulièrement gênant lorsque l'on doit avoir recours à plusieurs échanges entre le terminal et le serveur d'autant plus que la bande passante allouée pour la communication entre le serveur et le terminal (notamment mobile) peut être restreinte.

La présente invention propose un mécanisme de stockage de données propres à des objets destinés à être agencés au sein de pages multimédia. Plus particulièrement, l'invention propose un processus de transmission et de décodage de fonctions de «cache» (c'est-à-dire une anticipation de lecture suivie d'une mémorisation) des données d'objets, préalablement à l'agencement spatio-temporel de ces objets dans une page multimédia.

Pour ce qui concerne l'édition de scènes graphiques, plusieurs formats de représentation graphique d'animations graphiques existent actuellement. Cependant, aucun de ces formats ne propose un mécanisme de "cache" d'informations

sur les objets graphiques qui sont décrits au sein de la représentation graphique.

On connaît des techniques qui permettent de gérer des
5 informations de "cache" d'une scène graphique en utilisant
des méthodes programmatiques (par exemple en MPEG/MPEGJ,
ou VRML/EAI), ou encore protocolaires (par exemple en
http, RTP, ou autres), mais dans un but tout différent de
celui de la présente invention. D'ailleurs, ces techniques
10 souffrent d'un manque de souplesse en ce sens qu'il est
impossible de télécharger des morceaux de contenu
programmatique, ou encore parce que de telles techniques
utilisent nécessairement les mécanismes protocolaires
précités. Elles souffrent aussi d'un manque d'efficacité
15 dans le rendu graphique en ce sens qu'il est souvent
nécessaire de lancer une machine virtuelle pour traiter le
contenu programmatique.

L'un des buts visés par la présente invention concerne la
20 réduction de la mémoire des terminaux, nécessaire à
l'édition de pages multimédia complexes, ou d'une
succession de telles pages.

Un autre but visé par la présente invention concerne la
25 réduction des ressources de calcul nécessaires à l'édition
de telles pages, ou d'une succession de ces pages.

Un autre but visé par la présente invention est de fournir
un procédé permettant d'accomplir les buts visés ci-avant
30 tout en offrant une compatibilité avec les techniques
classiques de décodage.

Plus généralement, un but visé par la présente invention est d'offrir une plus grande flexibilité au niveau des requêtes et données échangées entre le serveur et le terminal.

La présente invention propose tout d'abord un procédé d'édition de pages multimédia auprès d'un terminal, dans lequel un serveur distribue, à un ou plusieurs terminaux, une partie au moins des pages multimédia sous forme d'instructions d'agencements d'objets au sein d'une page multimédia à construire, le procédé comportant :

- a) au moins une étape préalable pendant laquelle le serveur transmet des données relatives à au moins un objet à agencer dans une page multimédia à construire, avec une instruction de stockage desdites données, identifiées par un lien, dans une mémoire du terminal,
- b) et au moins une étape courante pendant laquelle le serveur transmet un fichier descriptif comportant ledit lien, pour éditer, auprès du terminal, ledit objet dans au moins une page multimédia en construction, par lecture des données stockées dans la mémoire du terminal.

Ainsi, le procédé au sens de l'invention offre une plus grande flexibilité au niveau des requêtes et données échangées entre le serveur et le terminal, en permettant une anticipation sur le ou les objets qui seront édités dans les pages multimédia par le terminal. Typiquement, l'étape b) seule peut être réitérée plusieurs fois pour

l'édition du même objet dans plusieurs pages multimédia successives, avantageusement sans télécharger nécessairement du serveur les données relatives à l'objet pour chaque page multimédia à construire.

5

Dans une réalisation, le serveur transmet au terminal lesdites données et l'instruction de stockage dans au moins un paquet de données.

10

Le stockage dans la mémoire du terminal est préférentiellement temporaire et l'instruction de stockage précitée comporte, dans cette réalisation, un paramètre de temporisation de valeur prédéterminée définissant une date d'expiration à compter de sa réception auprès du terminal.

15

Dans une autre réalisation, l'instruction précitée est de type "cache" entre le serveur et le terminal, avec une zone mémoire du terminal allouée pour le stockage des données associées à cette instruction, afin d'accélérer une exécution ultérieure répétée de l'étape b), pour le même objet.

20

Dans une réalisation, l'instruction de stockage comporte une information relative à un nombre de données que le terminal doit stocker en mémoire. Par exemple, l'instruction de stockage peut comporter un marqueur de début d'enregistrement des données et un marqueur de fin d'enregistrement des données, de sorte que le nombre de données précité est défini par ce couple de marqueurs.

25

30

Avantageusement, le fichier descriptif précité comporte un script incluant un ou plusieurs liens sous forme de requêtes URL (pour "UNIVERSAL RESSOURCE LOCATOR").

- 5 De façon avantageuse, l'instruction de stockage est réalisée sous la forme d'une commande correspondant à une fonction de bas-niveau.

On indique d'ailleurs que de nombreuses scènes graphiques
10 utilisent une représentation sous forme d'une liste de primitives (fonctions uniques d'exécution simple et dites de "bas-niveau"). Les fonctions de description de "cache" permettent alors d'anticiper la transmission et la lecture de plusieurs scènes graphiques au sein d'un service
15 multimédia composite. Cette représentation bas-niveau des fonctions de "cache" permet d'avoir une interaction fine avec les objets de l'animation tout en assurant un transport binaire entre le serveur et le terminal, notamment mobile. On réduit ainsi le temps de latence pour
20 l'utilisateur final du terminal.

Dans un mode de réalisation, l'objet est un objet de type graphique comportant au moins l'un des éléments parmi :

- une image,
- 25 - une séquence d'images,
- une séquence d'images synthétiques 2D,
- et une séquence d'images synthétiques 3D.

On indique que de telles séquences d'images sont
30 susceptibles d'être utilisées par exemple par le standard MPEG-4, pour l'édition de scènes graphiques.

Dès lors que cette instruction de stockage apparaît comme un moyen essentiel pour mettre en œuvre le procédé ci-avant, la présente invention vise aussi un produit
5 programme sous forme de code informatique, et comportant une instruction de stockage, dans une mémoire d'un terminal, de données relatives à un objet destiné à être édité, auprès dudit terminal, dans une ou plusieurs pages multimédia dans lesquelles l'objet est susceptible
10 d'intervenir. La présente invention vise aussi un signal comportant ce code. Ce signal et/ou le produit programme lui-même, peuvent être transmis du serveur au terminal, ou encore émaner d'un support mémoire qui coopère avec un lecteur du terminal (tel qu'un lecteur de CD-ROM ou
15 autre).

D'autres caractéristiques et avantages de l'invention apparaîtront à l'examen de la description détaillée ci-après, et des dessins annexés sur lesquels :

- 20 - la figure 1 illustre les échanges entre un serveur SER et un terminal TER, pour le déroulement des étapes du procédé au sens de l'invention,
- la figure 2 représente schématiquement et partiellement les éléments d'un terminal TER, et
- 25 - la figure 3 représente des agents logiciels interagissant pour l'édition de pages multimédia auprès du terminal TER.

En annexe, on a retranscrit un code informatique de la
30 commande "CACHE" (en format binaire) correspondant à l'instruction de stockage précitée. Il faut comprendre en

particulier que la description et son annexe présentent des caractéristiques susceptibles de contribuer à la définition de l'invention.

- 5 En se référant à la figure 1, le contexte d'application de l'invention peut être décrit par les étapes suivantes :
- Le terminal mobile TER demande une ou plusieurs pages multimédia, par exemple un contenu d'animation graphique, à un serveur SER (étape 11).
 - 10 - Le serveur SER renvoie un contenu (étape 12) comportant des données (audio, vidéo, texte, ou autres) propres à un objet, par exemple un objet graphique à éditer dans une scène graphique. Dans ce contenu est décrite une fonction de stockage temporaire "CACHE" (correspondant à l'instruction de stockage précitée), comme le montre
15 la table 12-a. Cette fonction "CACHE" indique qu'un ensemble d'objets graphiques sera stocké en mémoire dans le terminal et sera, le cas échéant, accessible en réponse à une requête du même terminal. Cette fonction
20 indique donc au terminal TER qu'il doit stocker des données relatives à un objet susceptible d'intervenir dans une ou plusieurs futures scènes graphiques à construire. Cet objet est identifié par un nœud graphique "node i" auquel on associe donc des données
25 DATA propres à l'objet graphique.
 - Le terminal stocke ces données DATA (étape 13) dans une zone mémoire ZMi du terminal, identifiable par un lien, par exemple une chaîne de caractère "node i".
 - Le serveur SER envoie (étape 15) la description d'une
30 scène incorporant l'objet "node i", le cas échéant avec des attributs Attr d'agencement de l'objet dans la

scène (fichier descriptif par exemple sous forme de script 15-b).

- Le terminal TER reconnaît le lien "node i" vers la zone mémoire ZMi et récupère (étape 16) les données DATA préalablement stockées à l'étape 13 pour éditer une scène graphique incluant l'objet correspondant au nœud "node i", avec les attributs d'agencement Attr qu'a transmis le serveur à l'étape 15.
- Le cas échéant, le terminal peut demander au serveur de nouveaux attributs, pour le même objet "node i", pour l'édition d'une nouvelle scène graphique incluant ce même objet (étape 14), si la durée de vie du stockage des données DATA auprès du terminal le permet, comme on le verra plus loin.

Cette commande CACHE est utilisée pour stocker temporairement un ensemble de données DATA propres à un objet d'une scène à construire. Cette commande est préférentiellement regroupée avec les données DATA dans un même paquet (par exemple un paquet AccessUnit en MPEG-4/System, ou encore un paquet RTP). Pour modifier une scène actuelle, le serveur transmet des paquets suivants (contenant une ou plusieurs commandes).

On se réfère à la figure 2 pour décrire brièvement des modules prévus classiquement dans un terminal TER. Le terminal comporte un module de communication 21, notamment avec le serveur SER, duquel il reçoit la commande "CACHE" précitée. A cette commande CACHE est associé un nœud graphique "node i" et le terminal stocke les données DATA dans une zone mémoire ZMi ($i=1,2,\dots$) en correspondance de

ce nœud "node i". Cette zone mémoire ZMi est prévue dans une mémoire MEM que comporte le terminal TER.

Ensuite, une commande suivante que transmet le serveur et relative à l'objet identifié par le nœud "node i", sera interprétée par le terminal comme visant le contenu de la zone mémoire ZMi et permettra donc de récupérer les données DATA relatives à cet objet. Ces données DATA sont ensuite traitées dans une mémoire de travail 22 (par exemple par un logiciel du terminal dit "PLAYER"), éventuellement distincte de la mémoire MEM. Les informations permettant l'édition de la scène à construire sont alors transférées vers une interface 23, par exemple une interface graphique pour un affichage de la scène (ou encore par une interface sonore pour un rendu sonore). Comme ces données DATA sont préférentiellement temporaires, il sera fait référence à la valeur horaire lue auprès d'une horloge 24, habituellement prévue dans un terminal (notamment dans son processeur).

On décrit ci-après la sémantique de la commande "CACHE" (décrite en format binaire dans l'annexe et appelée *cacheObject*).

Un paquet "cacheObject" permet d'envoyer en avance un contenu qui s'exécutera par exemple en réponse à une requête URL (par exemple à la suite des étapes 14 et 15 de la figure 1). Cette requête peut être faite suite à une commande d'un utilisateur du terminal (commande interactive). Un cacheObject peut être permanent, dans sa définition, et stocké en mémoire à n'importe quel instant.

Toutefois, les données associées au cacheObject ont une date d'expiration (dans la mémoire du terminal) définie par le temps de réception et la durée de vie TTL (pour "Time To Live") de ses données stockées temporairement.

5 Cet attribut TTL indique en secondes le temps après lequel le cacheObject n'est plus valide.

Les attributs qui sont associés à cette commande cacheObject sont par exemple les suivants :

- 10
- TTL : la durée de validité du cacheObject
 - URL : les données du cacheObject sont récupérées de la mémoire MEM, en réponse à ce lien URL
 - Source : les données DATA à stocker et qui pourront être récupérées ensuite.

15

Pour ce qui concerne les attributs Attr qui sont ensuite envoyés avec le nœud "node i", on cite à titre d'exemple une couleur et une police de caractères associées à un texte à afficher auprès du terminal, tandis que la chaîne

20 de caractères, elle-même, du texte à afficher a été stockée préalablement en mémoire MEM du terminal, en tant que données DATA envoyées avec la commande CACHE au sens de l'invention.

25 On indique que la plupart des scènes graphiques nécessitent habituellement une représentation sous forme d'une liste de primitives (fonctions de bas-niveau) de rendu graphique. A chacune de ces primitives correspond une seule fonction, simple, d'attribution d'un paramètre

30 d'édition graphique. La fonction de stockage CACHE, au sens de l'invention, se présente avantageusement comme une

fonction de bas-niveau. Une représentation bas-niveau de cette fonction permet notamment d'avoir une interaction fine avec les objets de l'animation et un transport binaire entre le serveur et le terminal, notamment mobile.

5

Plus spécifiquement, on a retranscrit en annexe jointe l'instruction "cacheObject", en langage SDL (pour "Synthetic Description Language"), correspondant à un code adopté pour définir les formats de *bitstreams* (flux binaire), avec des octets associés à chaque champ. On indique que des détails relatifs à ce langage informatique peuvent être trouvés dans le descriptif de la norme ISO IEC 144-96.

On indique ici que la commande CACHE est déclarée par l'instruction `const bit(4) 12` qui signifie que cette commande est déclarée par l'entier "12" qui devra se retrouver dans les quatre premiers bits du flux binaire que reçoit le terminal. Ainsi, si ces quatre premiers bits comportent l'entier "12", ils déclareront la commande CACHE. Ensuite, la commande `bit(6) msgID` déclare l'entier msgID qui, en fonction de sa valeur, lorsqu'il est reçu par le terminal, peut déclencher différentes actions auprès du terminal. Ici, ces actions sont "Permanent" (indique si le terminal doit garder les données de l'objet en mémoire ou non) et "Replace" (indique si le terminal doit remplacer le contenu de ses données mémorisées par les nouvelles données arrivantes). Des combinaisons de ces actions (ou non-actions) sont aussi prévues, selon la valeur de msgID.

30

Les commandes suivantes uint(X) déclarent :

- le nombre d'octets à lire et à stocker "payloadLength",
- la durée de vie des données stockées en mémoire "TTL",
et
- 5 - un temps à lire "time", auprès d'une horloge du
terminal 24 (figure 2), pour respecter la durée de vie
TTL.

La commande byte(8) [payloadLength] payload indique que le
terminal doit lire et, le cas échéant (selon la valeur de
10 la variable msgID), enregistrer toutes les données qu'il
reçoit du serveur jusqu'à ce qu'il lise le marqueur
"payloadLength".

Enfin, la commande suivante STZString définit une manière
15 de lire une chaîne de caractères pour lire un lien URL
soit vers la mémoire du terminal, soit vers des pages html
du serveur.

Ainsi, le serveur transmet des pages multimédia qui
20 contiennent des liens et les réponses associées à ces
liens avec une instruction de stockage de ces réponses sur
le terminal. Le terminal peut soit accepter, soit refuser
de stocker ces réponses, par exemple si elles sont trop
volumineuses ou si le délai d'expiration est trop court ou
25 trop long. Suite à une interaction de l'utilisateur ou
l'exécution d'un script pour une demande du contenu d'un
lien dans une des pages multimédia, le terminal accède
prioritairement à la réponse stockée sur le terminal, si
elle est effectivement stockée.

On se réfère maintenant à la figure 3 pour décrire un modèle de transmission et de rendu de scènes graphiques.

Une pluralité de modules 42 (décodeurs d'images divers),
5 43 (gestion de protocoles réseaux), 44 (gestion de polices (ou caractères) de texte), sont stockés en mémoire 41 du terminal TER, en tant que résidents mémoire. En outre, un logiciel client 45, dit "PLAYER", est aussi stocké en tant que résident mémoire dans la mémoire 41.

10

Le PLAYER 45 permet de visualiser des contenus animés, interactifs et multimédia sur le terminal mobile. Essentiellement, ce logiciel télécharge ou lit des informations qui décrivent l'agencement spatio-temporel
15 d'objets graphiques, la façon dont ils sont synchronisés et les interactions de l'utilisateur du terminal qui sont possibles sur ce contenu.

Le PLAYER 45 interprète alors les interactions de
20 l'utilisateur et en déduit les comportements appropriés des objets graphiques ou les requêtes à effectuer sur le serveur de contenus. Le PLAYER 45 (par exemple de type "STREAMEZZO") inclut des fonctions de rendu d'objets graphiques et des moteurs pour la visualisation (moteur de
25 rendu graphique 50) et l'interaction (moteur d'interaction 51) avec la scène multimédia. Le PLAYER 45 utilise les modules 42, 43, 44 en tant qu'API ("Application Program Interface") du système du terminal mobile qui permettent :

- de décoder les images (module 42),
- 30 • de récupérer un flux venant du réseau ou d'une source locale (module 43), et

- de gérer l'affichage du texte et notamment les polices résidentes en standard dans le terminal mobile (module 44).

5 Les contenus pour l'édition comprennent des graphiques vectoriels animés, du son, de la vidéo et des interactions utilisateur. La visualisation de contenus multimédia interactifs, dans des environnements mobiles, nécessite habituellement l'utilisation des techniques de compression
10 afin d'assurer une mise à disposition efficace du contenu et une optimisation du rendu des objets graphiques composant ce contenu. Ce contenu peut alors être visualisé dans de bonnes conditions sur les terminaux mobiles. Les informations lues ou téléchargées par le PLAYER 45 sont
15 donc fortement compressées et le PLAYER doit donc décompresser ces données et les interpréter à la volée pour jouer le contenu.

Plus particulièrement, le PLAYER 45 comporte des modules
20 de décodage audio et vidéo, respectivement 46 et 47, un analyseur de flux décodé 49, un gestionnaire de média 48, un moteur de rendu (graphique ou sonore) 50 et un moteur d'interactivité 51.

25 Le fonctionnement du logiciel PLAYER peut se décrire selon les étapes suivantes :

- saisie des données d'entrée via une connexion réseau ou une lecture de fichier,
- décompression de ces données afin d'obtenir une
30 description des objets graphiques directement

utilisables par le moteur de rendu audio et graphique
50,

- composition des objets graphiques entre eux pour créer une scène graphique,
- 5 - rendu graphique proprement dit des objets audio et graphiques, par affichage d'objets visuels ou par jeu d'un son,
- prise en compte des interactions utilisateurs, par exemple un clique d'un pointeur, ou la pression d'une
- 10 touche, ou autres,
- établissement d'une connexion à une source d'information locale ou distante si nécessaire.

Suite à une requête de l'utilisateur, cette dernière étape
15 va consister en l'ouverture d'une connexion vers le serveur SER et récupérer un flux binaire. Ce flux binaire est analysé par le PLAYER 45 qui crée alors un objet *ScèneGraph* contenant les différents objets de la scène sous forme de nœuds d'un graphe. Le flux d'information est

20 découpé en paquets qui regroupent des informations qui, dans une réalisation préférée, ne sont valides uniquement qu'à un instant donné et correspondent à un seul type d'information (principe de l'"AccessUnit" du standard MPEG-4).

25 Le PLAYER analyse chaque paquet et exécute les commandes qui y sont décrites suivant son horloge (non représentée sur les figures), laquelle fournit le temps de la scène multimédia et, le cas échéant, la durée de validité TTL des

30 données DATA associées à une commande CACHE.

Ainsi, on comprendra que ce PLAYER 45, notamment avec les autres modules 42, 43 et 44 du terminal, peut assurer le processus de réception et de décodage d'une fonction de stockage CACHE de données d'objets graphiques intervenant dans des scènes graphiques, au sens de la présente invention.

Cette fonction de stockage permet de gérer la représentation d'un objet et/ou la modification de son agencement dans une scène, sans avoir à télécharger systématiquement les mêmes données de l'objet dans le terminal. Ainsi, cette fonction permet de lier plusieurs scènes graphiques en un service multimédia composite, avantageusement lorsque le même objet intervient fréquemment dans ces scènes.

Plus généralement, le procédé au sens de l'invention permet une réduction de la mémoire (par exemple la mémoire graphique) du terminal, puisque l'on ne mémorise que l'instruction de stockage de données sur les nœuds graphiques.

Le procédé au sens de l'invention permet aussi un gain dans l'utilisation des ressources de calcul, puisque, typiquement, l'utilisation d'un contenu programmatique tel que défini ci-avant induirait un net surcoût de calcul, au moins pour certaines animations. Avantageusement, le procédé au sens de l'invention, utilisant un mécanisme conforme à un processus classique de rendu de commande graphique, dans une réalisation préférée, est alors facile

à implémenter, en particulier pour un système à terminaux mobiles coopérant avec des réseaux cellulaires.

Le procédé au sens de l'invention offre aussi une
5 compatibilité avec les techniques classiques de décodage, puisque le procédé peut être mis en œuvre dans la plupart des dispositifs de rendu graphique.

On indique que le procédé de l'invention, très général,
10 peut s'appliquer à pratiquement toutes les descriptions d'animations graphiques actuelles telles que MPEG-4/BIFS, SVG, ou autres, dès lors qu'une application requiert la représentation des signaux qui la composent sous forme d'un agencement spatio-temporel d'objets graphiques.

Annexe

```
cacheObject {
    constbit(4) 12;
5    bit(6) msgID;
    if (msgID == 2) {
        isPermanent = false;
        isReplace = true;
    } else if (msgID == 4) {
10    isPermanent = true;
        isReplace = true;
    } else if (msgID == 6) {
        isPermanent = false;
        isReplace = false;
15    } else if (msgID == 7) {
        isPermanent = true;
        isReplace = false;
    }
    uint(24) payloadLength;
20    uint(32) time;
    byte(8)[payloadLength] payload;
    uint(32) TTL;
    STZString url;
}

25    STZString {
        uint(5) len;
        uint(len) slen;
        byte(8)[len] UTF-string;
30    }
```

Revendications

1. Procédé d'édition de pages multimédia auprès d'un terminal, dans lequel un serveur distribue, à un ou
5 plusieurs terminaux, une partie au moins desdites pages multimédia sous forme d'instructions d'agencements d'objets au sein d'une page multimédia à construire,
caractérisé en ce que le procédé comporte :
- a) au moins une étape préalable pendant laquelle le
10 serveur transmet des données relatives à au moins un objet à agencer dans une page multimédia à construire, avec une instruction de stockage desdites données, identifiées par un lien, dans une mémoire du terminal,
- b) et au moins une étape courante pendant laquelle le
15 serveur transmet un fichier descriptif comportant ledit lien, pour éditer, auprès du terminal, ledit objet dans au moins une page multimédia en construction, par lecture des données stockées dans la mémoire du terminal.
- 20 2. Procédé selon la revendication 1, caractérisé en ce que le serveur transmet au terminal lesdites données et l'instruction de stockage dans au moins un paquet de données.
- 25 3. Procédé selon l'une des revendications 1 et 2, caractérisé en ce que le stockage dans la mémoire du terminal est temporaire.
4. Procédé selon la revendication 3, caractérisé en ce que
30 ladite instruction comporte un paramètre de temporisation

de valeur prédéterminée définissant une date d'expiration à compter de sa réception auprès du terminal.

5 5. Procédé selon l'une des revendications précédentes, caractérisé en ce que ladite instruction est de type "cache" entre le serveur et le terminal, avec une zone mémoire du terminal allouée pour le stockage des données associées à ladite instruction, afin d'accélérer une exécution ultérieure répétée de l'étape b), pour le même
10 objet.

6. Procédé selon l'une des revendications précédentes, caractérisé en ce que l'instruction de stockage comporte une information relative à un nombre de données que le
15 terminal doit stocker en mémoire.

7. Procédé selon l'une des revendications précédentes, caractérisé en ce que ledit fichier descriptif comporte un script incluant un ou plusieurs liens sous forme de
20 requêtes URL.

8. Procédé selon l'une des revendications précédentes, caractérisé en ce que ladite instruction de stockage est réalisée sous la forme d'une commande correspondant à une
25 fonction de bas-niveau.

9. Procédé selon l'une des revendications précédentes, caractérisé en ce que le terminal est un terminal mobile agencé pour coopérer avec un réseau cellulaire.

10. Procédé selon l'une des revendications précédentes, caractérisé en ce que ledit objet est un objet de type graphique comportant au moins l'un des éléments parmi :

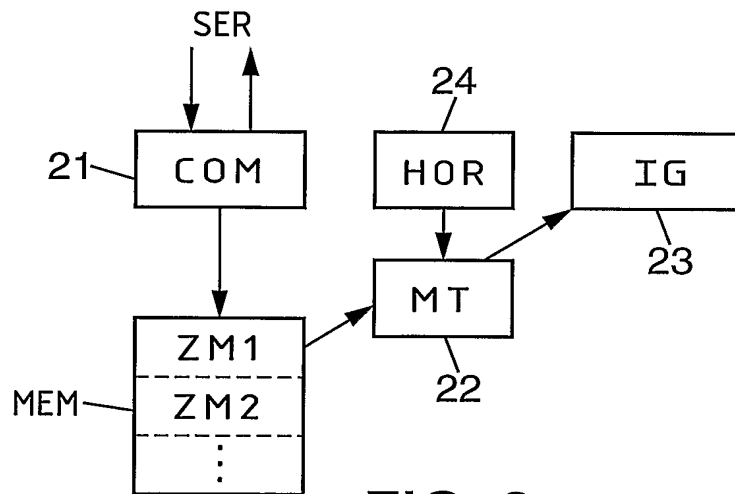
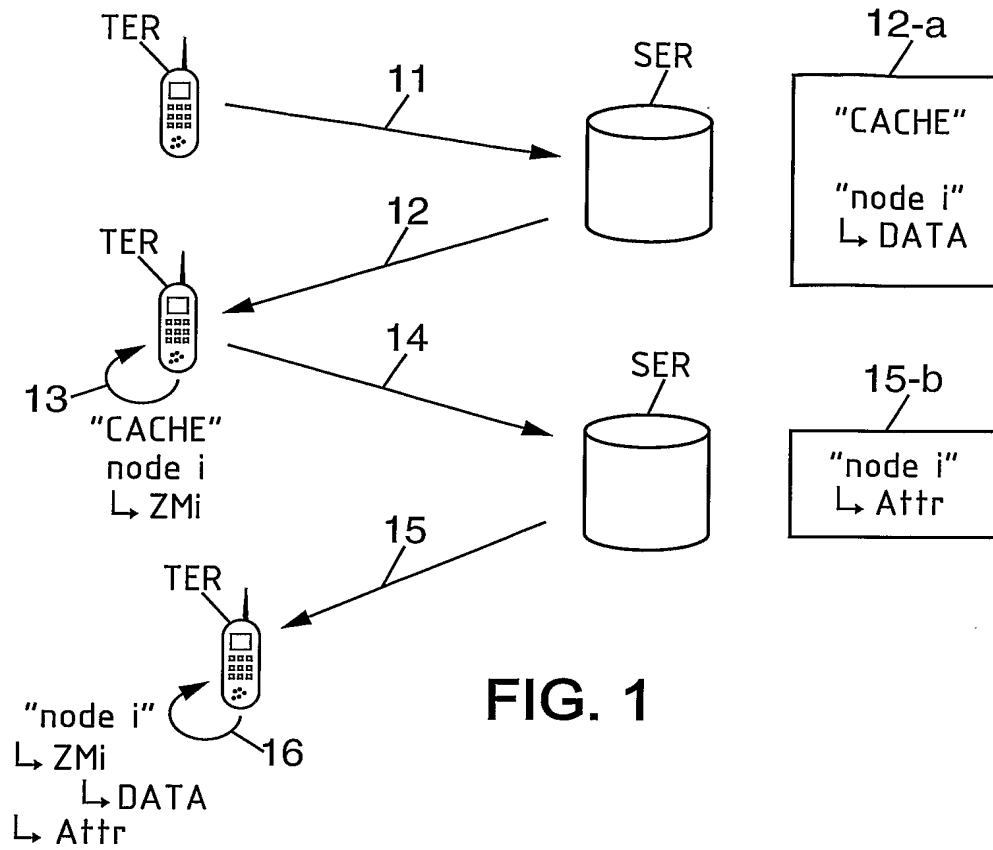
- une image,
- 5 - une séquence d'images,
- une séquence d'images synthétiques 2D,
- et une séquence d'images synthétiques 3D.

11. Produit programme sous forme de code informatique, caractérisé en ce qu'il comporte une instruction de
10 stockage, dans une mémoire d'un terminal, de données relatives à un objet destiné à être édité, auprès dudit terminal, dans une ou plusieurs pages multimédia dans lesquelles ledit objet est susceptible d'intervenir.

15

12. Signal comportant un code informatique, caractérisé en ce que le code comporte une instruction de stockage, dans une mémoire d'un terminal, de données relatives à un objet destiné à être édité, auprès dudit terminal, dans une ou
20 plusieurs pages multimédia dans lesquelles ledit objet est susceptible d'intervenir.

1/2



2/2

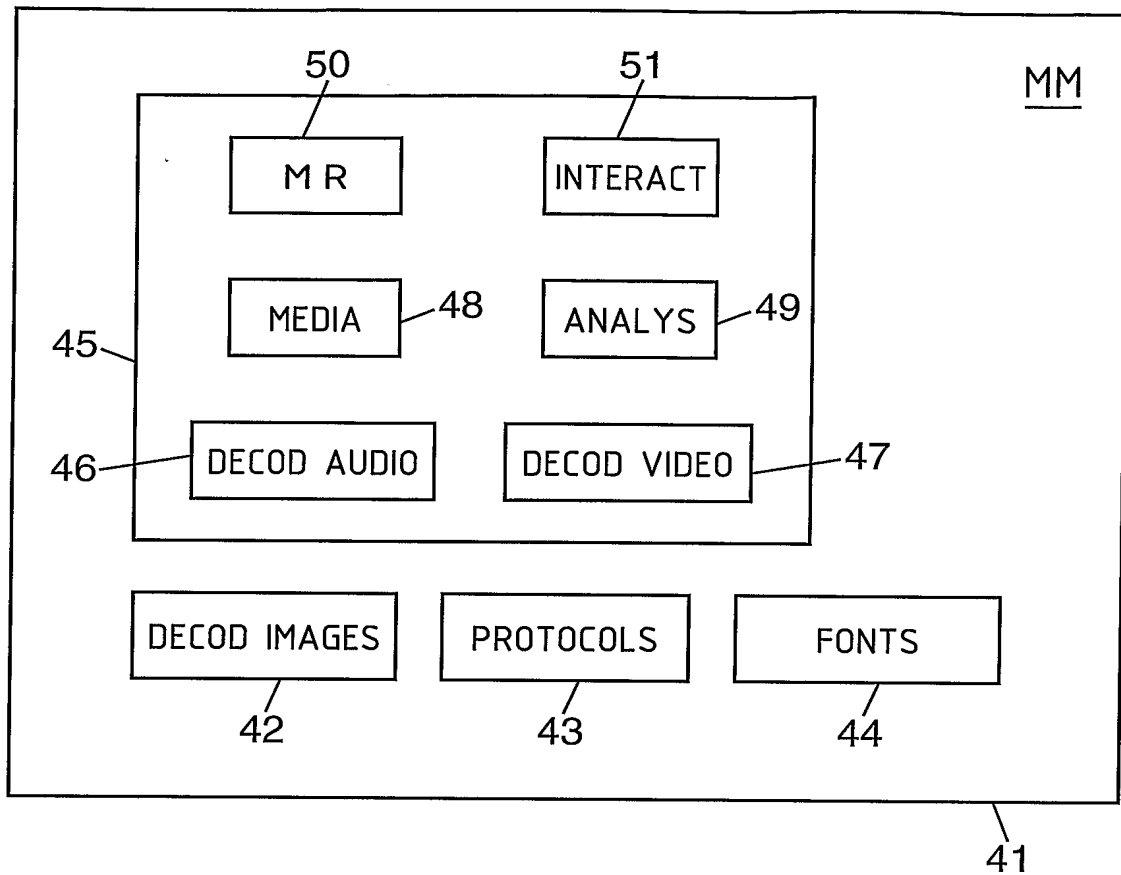


FIG. 3

INTERNATIONAL SEARCH REPORT

International Application No

PCT/FR2004/000337

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/152229 A1 (PENG LUOSHENG) 17 October 2002 (2002-10-17) abstract figures 2,5-8,13,14,16 paragraphs '0003!, '0005!, '0008!, '0041! - '0043!, '0046!, '0047!, '0061! -----	1-12
A	US 6 591 288 B1 (DRISCOLL RICHARD JOHN ET AL) 8 July 2003 (2003-07-08) the whole document -----	1-12
A	US 2004/030832 A1 (SQUIBBS ROBERT FRANCIS) 12 February 2004 (2004-02-12) the whole document -----	1-12



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents :

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search

13 October 2004

Date of mailing of the international search report

21/10/2004

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Pereira, M

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PT/FR2004/000337

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 2002152229	A1	17-10-2002	NONE	
US 6591288	B1	08-07-2003	NONE	
US 2004030832	A1	12-02-2004	EP 1388973 A1	11-02-2004
			EP 1388974 A1	11-02-2004
			GB 2391773 A	11-02-2004
			GB 2391626 A	11-02-2004
			GB 2391782 A	11-02-2004
			GB 2391759 A	11-02-2004
			GB 2391760 A	11-02-2004
			GB 2391761 A	11-02-2004
			GB 2391661 A	11-02-2004
			GB 2391662 A	11-02-2004
			GB 2392284 A	25-02-2004
			GB 2392274 A	25-02-2004
			GB 2392285 A	25-02-2004
			GB 2391663 A	11-02-2004
			US 2004093466 A1	13-05-2004
			US 2004030491 A1	12-02-2004
			US 2004030494 A1	12-02-2004
			US 2004132467 A1	08-07-2004
			US 2004097226 A1	20-05-2004
			US 2004097242 A1	20-05-2004
			US 2004137911 A1	15-07-2004

RAPPORT DE RECHERCHE INTERNATIONALE

Demande Internationale No

PCT/FR2004/000337

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 H04L29/06

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)

CIB 7 H04L

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	US 2002/152229 A1 (PENG LUOSHENG) 17 octobre 2002 (2002-10-17) abrégé figures 2,5-8,13,14,16 alinéas '0003!, '0005!, '0008!, '0041! - '0043!, '0046!, '0047!, '0061! -----	1-12
A	US 6 591 288 B1 (DRISCOLL RICHARD JOHN ET AL) 8 juillet 2003 (2003-07-08) le document en entier -----	1-12
A	US 2004/030832 A1 (SQUIBBS ROBERT FRANCIS) 12 février 2004 (2004-02-12) le document en entier -----	1-12

☐

Voir la suite du cadre C pour la fin de la liste des documents

☒

Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

- *A* document définissant l'état général de la technique, non considéré comme particulièrement pertinent
- *E* document antérieur, mais publié à la date de dépôt international ou après cette date
- *L* document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)
- *O* document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens
- *P* document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

- *T* document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention
- *X* document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
- *Y* document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier
- *Z* document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée

13 octobre 2004

Date d'expédition du présent rapport de recherche internationale

21/10/2004

Nom et adresse postale de l'administration chargée de la recherche internationale

Office Européen des Brevets, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Fonctionnaire autorisé

Pereira, M

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements rela

ix membres de familles de brevets

Demande Internationale No

FR2004/000337

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2002152229	A1	17-10-2002	AUCUN	
US 6591288	B1	08-07-2003	AUCUN	
US 2004030832	A1	12-02-2004	EP 1388973 A1	11-02-2004
			EP 1388974 A1	11-02-2004
			GB 2391773 A	11-02-2004
			GB 2391626 A	11-02-2004
			GB 2391782 A	11-02-2004
			GB 2391759 A	11-02-2004
			GB 2391760 A	11-02-2004
			GB 2391761 A	11-02-2004
			GB 2391661 A	11-02-2004
			GB 2391662 A	11-02-2004
			GB 2392284 A	25-02-2004
			GB 2392274 A	25-02-2004
			GB 2392285 A	25-02-2004
			GB 2391663 A	11-02-2004
			US 2004093466 A1	13-05-2004
			US 2004030491 A1	12-02-2004
			US 2004030494 A1	12-02-2004
			US 2004132467 A1	08-07-2004
			US 2004097226 A1	20-05-2004
			US 2004097242 A1	20-05-2004
			US 2004137911 A1	15-07-2004